Microsoft Exchange Server

Chapter 9
Upgrading Other Messaging Components

Deployment Guide

Abstract

This chapter outlines the steps for upgrading other messaging components such as: Key Management Service, Chat service, Event Service and third party software in general.

This is a preview edition of the Microsoft Exchange 2000 Server Upgrade Series. The final documentation will be available in November 2000.

© 2000 Microsoft Corporation. All rights reserved.

THIS IS PRELIMINARY DOCUMENTATION. The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft, Active Directory, Windows, Windows NT and the Windows logo are registered trademarks of Microsoft Corporation.

Other product or company names mentioned herein may be the trademarks of their respective owners.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

1000

CONTENTS

INTRODUCTION.....1

MICROSOFT KEY MANAGEMENT SERVICE (KMS.....2
Introduction to PKI and Advance Security using KMS 2
Establishing a Trust Hierarchy 2
Exchange 2000 Security Features 2
Certificate Services and Exchange 2000 KMS 3
Deployment Strategy 4
The Active Directory 4
Client functions 5
Upgrading Exchange 4.0 and 5.x to Exchange 2000 Server 5
Upgrading the Public Key Infrastructure to Windows 2000 5
Next Step: Upgrade to Windows 2000 SP1 7
Convert the Certificate Servers From Stand-Alone CAs to Enterprise CAs 7
Add Templates to the Issuing CA Server 8
To install KMS 9
KM Server Forklift Upgrade Process 9
Key Management Service Passwords 10
Exchange Administrator (KMS) functions 11
If Things Go Wrong 12
Exchange 2000 KMS Bulk Import/Export 13
Archive database details 15
Things to know 15
Additional References 16

CHAT SERVICES.....18
Introduction to Chat Services 18
Migrating from Exchange 5.5 Chat Service 18
Chat Service Migration 19
If Things Go Wrong 19

EVENT – SCRIPT SERVICES.....20
Introduction to Microsoft Exchange Server Scripting and Routing 20
Microsoft Exchange Server Scripting Agent 20
Microsoft Exchange 5.5 Server Routing 20
Event Support in Exchange 2000 21
Upgrading your Event-Script Services 21
If Things Go Wrong 21

UPGRADING THIRD PARTY SOFTWARE.....22
Introduction 22
If Things Go Wrong 22

SUMMARY.....23

TERMINOLOGY....24
Key Management Services 24
Encryption 24
Hash Functions 25
Ciphers 25
Secure Transmission 26
X.509v1 Key Exchange (Item Encipherment and Signing) 26
X.509v3 Key Exchange (Item Encipherment and Signing) 26
Certificate Services and the Key Management Service 27
Event Script Services 29
The Exchange Event Scripting Agent 30
Microsoft Exchange Server Routing 30

MICROSOFT KEY MANAGEMENT SERVICE (KMS

This chapter discusses the upgrading of the other messaging services such as Key Management Services, Chat Services, Event-Script Services and third party software in general. It is important to emphasize that upgrading these services need very careful planning before you actually start your upgrade to Windows 2000 and Exchange 2000.

- KMS is a security tool in Exchange 5.5 that protects data integrity through message encryption and digital signatures. It also sets up a centralized management of keys and certificates for all enrolled users.
- Chat Services are provided through Exchange 5.5 as an add-on to an existing messaging infrastructure. Chat services provide virtual meeting places for scheduled or spontaneous text-based discussions.
- Exchange 5.5 uses the Scripting Agent and Routing Capabilities to turn a public folder into an application, enabling it to run event-triggered server side scripts and control the flow of the messages or forms in Exchange Organization.
- Many organizations use third party software like fax connectors, backup software, monitoring tools, and anti-virus tools to add extra functionality to Exchange 5.5. The general rules for upgrading such software are described at the end of this chapter.

Introduction to PKI and Advance Security using KMS

When deploying Advanced Security, first determine the organizational topology. Locate the KM Server(s) in a secure but accessible spot. Location becomes more important when you deploy Exchange 2000 with multiple Certificate Servers and KM Servers. A smart placement plan for CSs and KM Servers can simplify backup and recovery of KM databases and the key issuance policies.

Next, deploy the Exchange servers. Only S/MIME clients (Outlook 98 and above) clients should be deployed. Since Exchange 2000 uses Certificate Server (CS) to generate public keys and certificates, a CS must already be deployed in the architecture. When deployment is complete, you can decide the client functionality. If you want to sign and secure items within the organization and are not concerned about compliance with S/MIME standards, you can deploy any of the Outlook or Exchange clients.

Starting with Exchange 5.5SP1, Exchange 5.5 provided a new server-side feature giving non-S/MIME clients two options for handling S/MIME

signatures: *Preserve* and *Toss*. “Preserve S/MIME signature” stores the entire signed message in an attachment to the message. Messages go through a conversion process when they are received into the Exchange store: if even one bit is changed during this process, the signature is invalid, even though the message may not have changed. Using the preserve option and encapsulating the original mail message as an attachment preserves the signature on the item. “Toss S/MIME signature” removes the signature associated with the message. Using this method, S/MIME clients lose the ability to verify signatures on messages, but all clients get readable text.

Establishing a Trust Hierarchy

When the supporting architecture is in place, you can set up a trust network with external organizations.

First, decide which external organizations or Certificate Authorities (CAs) your organization should trust, and then obtain their X509v3 root certificates. Conduct this crucial transfer out-of-band. With the CAs in hand, you can populate the Certificate Trust Lists (CTL) property page, which directories access to verify certificates. Note that this process is performed on each KMS, not each CS.

Once the trust is established, you should regularly import CRLs for trusted external CAs (and export yours to them), to guarantee that revocations are reflected in the Certificate Revocation Lists (CRL). When other organizations establish trust with yours, they will request your root certificate and CRL. You can obtain these by exporting them to a file from your CS. As before, transmit these out-of-band.

Exchange 2000 Security Features

Exchange 2000 Server provides security features for securing messages specifically. These features can secure messaging between users, help deter unsolicited email, separate administrative abilities in your organization, and give users secure methods of communicating with Exchange 2000 Server. The main security features of Exchange 2000 are:

- Key Management Service
- Virtual server security
- Permissions
- Securing client and server communication

Certificate Services and Exchange 2000 KMS

In addition to exploiting the built-in security of Windows 2000 Server, Exchange 2000 Server provides advanced security in the form of an optional component named the Key Management Service (KMS). The KMS

works in conjunction with the Active Directory to manage the encryption keys used to encrypt e-mail messages. It integrates with Microsoft Certificate Services to provide services.

The Exchange KMS is the communication hub between the Certificate Service, the Administrator, the Exchange-specific Crypto Service Provider (CSP), the System Attendant, and the Archive Database. It is the core component to Exchange Advanced Security, which recognizes and coordinates the flow of all tasks such as enrollment, recovery, renewal, and others.

The Exchange 2000 KMS architecture is the same as it was in Exchange 5.5. KMS still uses a private JET database (kmsdir.edb) to store the KMS administrator accounts and passwords as well as users' private-encryption-key and encryption-certificate histories.

Exchange 2000 requires Windows 2000 Active Directory and a Windows 2000 Certificate Sever. Active Directory stores user certificates, CRLs, and encryption preferences.

Exchange uses several components to secure items:

- **Exchange System Manager**—manages the Advanced Security operations for an organization, including issuance and revocation of certificates, recovery of certificates, and setting of site encipherment algorithm preferences.
- **KMS Archive database**—stores private encrypting keys.
- **Windows 2000 Certificate Server**—Windows 2000 Certificate server is required to run KMS.
- **Active** Directory—Generates private signing and encrypting keys, and stores certificates/preferences for client encipherment and signing. For example, clients state the maximum strength security that they support (for example, 128-bit encryption) in the directory. Other clients use this information to secure messages in ways they will be able to decipher.
- **Client (Outlook)**—the client interface to Advanced Security. It reads from te Active Directory users' certificates, performs all cryptography on mail messages, and protects private keys in a secure store.

The Enroll Agent (Computer), Exchange User, and Exchange User Signature templates of the CA must be enabled before KMS installs.

KMS can be configured to require more than one administrator to be present before performing each KMS task. Each task is initially configured to require only one password, and the first password is always "password" until it is changed.

KMS implements a dual-key architecture, which gives each enrolled user two key pairs and certificates, one used for encryption, and the other used for signing. KMS uses this architecture so the signing key does not need to be archived in KMS and enforces non-repudiation.

Earlier versions of KMS restricted KMS to just one server per site. However, Exchange 2000 KMS fits the site-based model. The KMS organizational unit becomes the Administrator Group and each newly created administrator group allows KMS to install on, or point to, an existing KMS server. You can also enroll users in bulk by enrolling selected users, groups, or servers.

When you implement encryption on the application layer, content filtering or virus checking across gateways is compromised.

Deployment Strategy

Networking KM Servers can be complex and risky because users can lose the ability to recover their key history if networking is improperly done. Plan to move to a X.509v3-based environment carefully. One prudent strategy is to upgrade all existing servers to Exchange Server 5.5 SP3, and then plan any alterations to the KM Server infrastructure as a separate project.

Both Microsoft Consulting Services (MCS) and Microsoft Product Support Services (PSS) can be instrumental in helping an organization plan a new Key Management infrastructure; these groups should be consulted as appropriate.

The Active Directory

Active Directory in Windows 2000 replaces the Security Accounts Manager (SAM) in Windows NT Server 4 as the security database. However, like an object in the SAM, each Active Directory object is given a 96-digit, pseudorandom Security Identifier (SID) that is globally unique.

Not all objects in Active Directory are assigned an SID. For instance, a security group has an SID, but a distribution group does not. Likewise, mail-enabled users have SIDs, but mail-enabled contacts do not. Only those objects that have SIDs can be added to the access control list (ACL) of a resource. If an object does not have a SID, it cannot be placed in the ACL. Therefore, non-SID objects cannot access resources guarded by an ACL.

Public information is stored in and retrieved from Active Directory. The directory also contains KM Server information clients need to use advanced security such as: X.509 v3 public signing and data encipherment certificates for each enrolled user. Publication of these certificates allows clients to look each other up in the Directory (rather than exchange keys manually).

Exchange 2000 allows you to store cross certificates of other certifying authorities in the directory. Publication of these certificates allows clients to authenticate and trust the root certificate of external organizations and other CAs within their organization. For example, if Microsoft and Company A both use the same CA, the establishment of a trust between Microsoft and Company A would be facilitated because initial identity negotiation would not be necessary: both organizations are automatically valid.

CRLs for both the home organization and cross-certified organizations must be public in the Directory so clients can determine if a message should be trusted.

Client functions

Enrolled clients can send/receive signed email, receive enciphered email, send enciphered email to enrolled recipients, and have KMS administrators recover their private key history. Clients generating secured and signed messages perform all cryptographic operations with local private, public, and public keys derived from Active Directory. Once enrollment has concluded for generating secure or signed messages, no traffic is generated between client and Key Management Server.

If you send a message to several users, the symmetric key for message encipherment or item signing is generated at the lowest level common to the recipients (value queried from Active Directory). Symmetric keys are generated to send a message and used only once. If someone derives a symmetric key by attacking a message, it is good only for that message.

Outlook 98 and Outlook 2000 provide a fully interoperable S/MIME client, using standard X509v3 certificates, and supporting the Exchange 4.0/5.0 format for backward compatibility. Outlook Express cannot address a KMS, so it obtains X.509v3 certificates from another CA and therefore does not support Exchange 4.0/5.0 format secured messages. Each time a client connects, Outlook reads the CRL from the directory to verify signatures.

Upgrading Exchange 4.0 and 5.x to Exchange 2000 Server

Several options are available for upgrading your existing Exchange servers to Exchange 2000. They are:

- Perform an in-place upgrade of your Exchange 5.5 servers.
- Install Exchange 2000 on new hardware and then migrate mailboxes and public folders to the new server.
- Use the leapfrog method, reusing hardware as the migration progresses throughout the organization.

Upgrading the Public Key Infrastructure to Windows 2000

In-place upgrade is supported if the KMS and CS are on the same

operating system. The underlying code is upgraded during Exchange Setup. When the KMS starts, the original KMS database is upgraded.

In Exchange 5.5, when users are deleted, their key history and certificates remain in the KMS database. In Exchange 2000, when users are removed from Active Directory, their key history is removed.

Do not install the Exchange 2000 version of KMS into a mixed–mode organization. Instead, continue using the existing Exchange 5.5 KMS and manage it from the Exchange 5.5 Administrator program.

For Exchange Server 5.5 KM server to use a Windows 2000 Server CA, both the KM server and the CA must be running the Windows 2000 Server operating system; if Exchange Server is deployed on Windows NT Server 4.0, Exchange Server cannot communicate with a Windows 2000 Server CA. The Windows 2000 Server CA can be installed on the same computer that KM server is installed on, or on a separate physical server. To facilitate this change in an existing deployment, you can upgrade a computer running a Windows NT Server 4.0 KM server and Microsoft Certificate Server version 1.0 to Windows 2000 Server, and set up Certificate Services.

If an organization does not want to upgrade the Exchange Server 5.5 KM server to Windows 2000 Server, that organization can use a Windows NT Server 4.0 Certificate Server only. The policy module that is installed for Exchange Server 5.5 KM server is an exclusive policy that ensures that the certificates the CS issues are marked for email [or Secure/Multipurpose Internet Mail Extensions (S/MIME)] use only. After this module is installed on any CA, that authority cannot service other types of certificate requests. Therefore, to support the KM server functions, a Windows 4.0 Server CA must be a stand-alone CA; it cannot be used as an enterprise CA.

Before you can install KMS on an Exchange 2000 server, you must install Windows 2000 Server in your Exchange organization. CS becomes your organization's CA, issuing X.509v3 certificates to your enrolled Advanced Security users.

When installing CS to use with KMS, Microsoft recommends changing the *Valid for* field in the CA Identifying Information screen to five years. This is the amount of time before your CA certificate expires. Your CA cannot issue certificates that are valid beyond the period of its own certificate expiration. For administrative purposes, it is better to have a CA certificate that is valid for at least five years.

Certificate Services also track and maintain CTLs and CRLs for your organization. Be sure to install the Enterprise version of Certificate Services. For more information on Certificate Services, see the Windows 2000 documentation.

Exchange 2000 KM servers are compatible with KM servers running Exchange 5.5 SP1. To upgrade an Exchange 5.5 KM server to Exchange 2000, follow the same procedure you would for any non-KMS Exchange server. For more information, see Chapter 5 on Upgrading Exchange 4.0 and 5.x to Exchange 2000 Server.

Next Step: Upgrade to Windows 2000 SP1

After upgrading the domain controller of the domain in which the certificate servers are set up with Windows 2000 and Active Directory, Windows NT 4.0 certificate servers are migrated to Windows 2000.

The CS databases needed to be converted to the new database format used in Windows 2000.

To do this, on all the certificate servers (root CA, policy CA, and issuing Key Management Server CA) execute the following commands to stop the certificate server, convert the database, and then start it again.

- net stop certsvc
- certutil –ConvertMDB
- net start certsvc

To prepare the certificate servers for use by Exchange 2000, they must be upgraded to enterprise CAs from the stand-alone CAs that existed in Windows NT 4.0.

After upgrading the root CA, policy CA, and issuing CA to enterprise CA computers, templates need to be added to the issuing CA so that Exchange 2000 Key Management Server can use the Certificate Server in the future.

Convert the Certificate Servers From Stand-Alone CAs to Enterprise CAs

Follow this procedure for each of the three CAs starting with the root CA, then the policy CA, and lastly the issuing CA.

1. Click **Start**, point to **Settings**, click **Control Panel**, double-click **Add/Remove Programs**, and then click **Add/Remove Windows Components** to remove Certificate Services.
2. Under **Components**, clear the **Certificate Services** check box and then click **Next**. When installation has completed, click **Finish**. The Windows Component Wizard will stop Certificate Services and remove it. However, the certificate services database is not removed.
3. Stop the **Microsft Exchange Chat** service if it is installed.
4. Click **Start**, point to **Settings**, click **Control Panel**, double-click **Add/Remove Programs**, and then click **Add/Remove Windows Components** to reinstall certificate services.
5. Under **Components**, select the **Certificate Services** check box. At the warning, click **Yes**, and then click **Next** to start the installation.
6. On the **Certification Authority Type** wizard page, under **Certification Authority types**, do one of the following:

    1. For the root CA, click **Enterprise root CA**

2. For the policy CA or issuing CA, click **Enterprise subordinate CA**.
7. Select the **Advanced options** check box, and then click **Next**:
8. On the **Public and Private Key Pair** wizard page, under **CSP**, click **Microsoft Base Cryptographic Provider v1.0**, select the **Use existing keys** check box, use the existing key for the old certificate server installation on the computer and, under **Hash algorithm**, click **SHA-1**. Make sure the **Use the associated certificate** check box is selected, and click **Next**.
9. On the **CA Identifying Information** wizard page, type a description for the CA, and then click **Next**.
10. On the **Data Storage Location** wizard page, select to store configuration information in the same shared folder that was set earlier, and select **Preserve existing certificate database**. Click **Next** to start the installation.
11. If you see a dialog asking if you want to stop services, click **OK**.
12. If the **Files Needed** dialog box appears, under **Copy files from**, type a path to the binaries that are needed to complete the installation, and then click **OK** to complete the installation.
13. On the **Completing the Windows Components Wizard** screen, click **Finish**.

Add Templates to the Issuing CA Server

14. Click **Start**, point to **Programs**, point to **Administrative Tools**, and click **Certificate Authority**.
15. In the console tree, expand the node for the issuing **Key Management Server CA**, right-click **Policy Settings**, point to **New** and then click **Certificate to Issue**.
16. In the **Select Certificate Template** dialog, hold down the Ctrl key and select **Enrollment Agent (Computer)**, **Exchange User**, and **Exchange Signature Only** policy templates, and then click **OK**. The chosen templates are installed and can be used by Exchange 2000.

The issuing CA can now be used by an Exchange 2000 installation. With Exchange 2000 the upgrade process to Exchange 2000 will upgrade the KM server installation and use this CS and the templates installed.

The following installation procedure is just one necessary step you must take to deploy KMS in your organization. The following table illustrates additional procedures that must be performed at different points during the KMS installation:

Procedure	When to Perform It
Install Exchange Certificate Templates	After installing Certificate Services, and before installing KMS. Enrollment agent (Computer) Exchange User Exchange Signature only
Select the Startup Password Location	During Exchange/KMS installation.
Grant Manage Rights to the Key Management Server	After KMS is installed, before requesting certificates from Certificate Services

To install KMS

1. Start the Microsoft Exchange 2000 Installation Wizard. In the **Component Selection** screen, note that Upgrade has been selected in the Action column and click **Next**.
2. On the **Service Account** page, enter the Exchange 5.5 service account password and click **Next**.
3. On the **Component Summary** page, click **Next**.
4. When installation is complete, click **Finish**.
5. If you want to install KMS on a computer currently running Exchange 2000, insert the Exchange CD and start the Microsoft Exchange 2000 Installation Wizard again. In the Component Selection window, select **Change** next to **Microsoft Exchange 2000** and then select **Change** next to **Microsoft Exchange Messaging and Collaboration Services**. Select **Install** next to **Microsoft Exchange Key Management Service**, and click **Next**.

KM Server Forklift Upgrade Process

If you want a Windows 2000 Server Certificate Server to issue X.509 V3 certificates to a computer running Windows 2000 Server and Exchange Server 5.5 Service Pack 3 KM server, you might need to upgrade from an existing Windows NT Server 4.0 environment. One method of migrating the Exchange Server 5.5 KM server database from Windows NT Server

4.0 to Windows 2000 Server is to perform a *forklift upgrade*. To perform a forklift upgrade, you copy the KM server database and log files from the computer running Windows NT Server 4.0 and Exchange Server 5.5 KM server, to a computer running Windows 2000 Server and Exchange Server 5.5 Service Pack 3 KM server. The computer running Windows 2000 Server and Exchange Server 5.5 Service Pack 3 KM server must have the same host name as the computer running Windows NT Server 4.0 and Exchange Server 5.5 KM server. To perform a forklift upgrade:

1. Copy the KM server database and log files from the computer running Windows NT Server 4.0 and Exchange Server 5.5 KM server to a shared network drive.
2. Disconnect the computer that is running Windows NT Server 4.0 and Exchange Server 5.5 KM server from the network.
3. Install Windows 2000 Server on the target computer. Make sure that you set up this computer using the same host and machine name as the computer that is running Windows NT Server 4.0 and Exchange Server 5.5 KM server.
4. Connect the computer running Windows 2000 Server to the subnet in which it will reside.
5. Set up Exchange Server 5.5 with KM server on the computer running Windows 2000 Server. Add the computer running Windows 2000 Server and Exchange Server 5.5 KM server to the same Exchange Server site that the computer running Windows NT Server 4.0 and Exchange Server 5.5 KM sever was previously a member of.
6. Install Exchange Server 5.5 Service Pack 3 on the computer running Windows 2000 Server and Exchange Server 5.5 KM server.
7. Shut down all of the Exchange Server services on the computer running Windows 2000 Server and Exchange Server 5.5 Service Pack 3 KM server.
8. Remove the KM server database and log files from the computer running Windows 2000 Server and Exchange Server 5.5 Service Pack 3 KM server.
9. Copy the KM server database and log files (originally from the computer running Windows NT Server 4.0 and Exchange Server 5.5 KM server) from the shared network drive that you copied them to in step 1, to the computer running Windows 2000 Server and Exchange Server 5.5 Service Pack 3 KM server.
10. Start all of the Exchange Server services on the computer running Windows 2000 Server and Exchange Server 5.5 Service Pack 3 KM server.

After you complete these steps, the Exchange Server 5.5 KM server database is migrated from the computer running Windows NT Server 4.0 to the computer running Windows 2000 Server and Exchange Server 5.5 Service Pack 3 KM server.

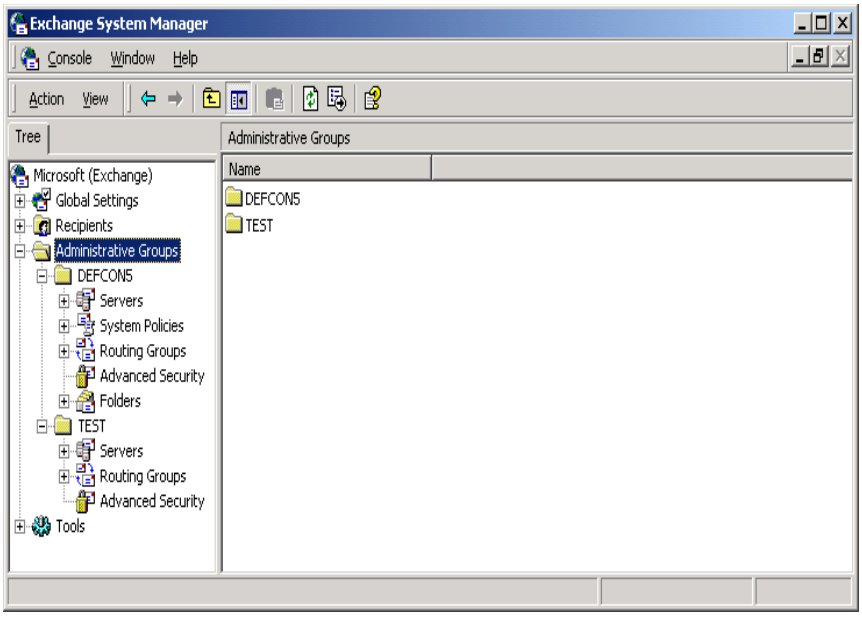
Key Management Service Passwords

The first security measure for protecting KMS is a given start-up parameter (password) to successfully start the KMS. For each restart of KMS, the startup parameter is mandatory and must be typed manually or read from a removable media. This startup parameter should be stored in a restricted location.

The second security measure is that administrative passwords are required to perform tasks in KMS. By default, only one administrative password is required to perform any task. Administrators, however, can set varying levels of security by requiring up to four different administrator passwords. Each task can be executed with approval of two or more KMS administrators. The most important management tasks are adding and removing KMS administrators and configuring the number of passwords required to execute each KMS functions.

Exchange Administrator (KMS) functions

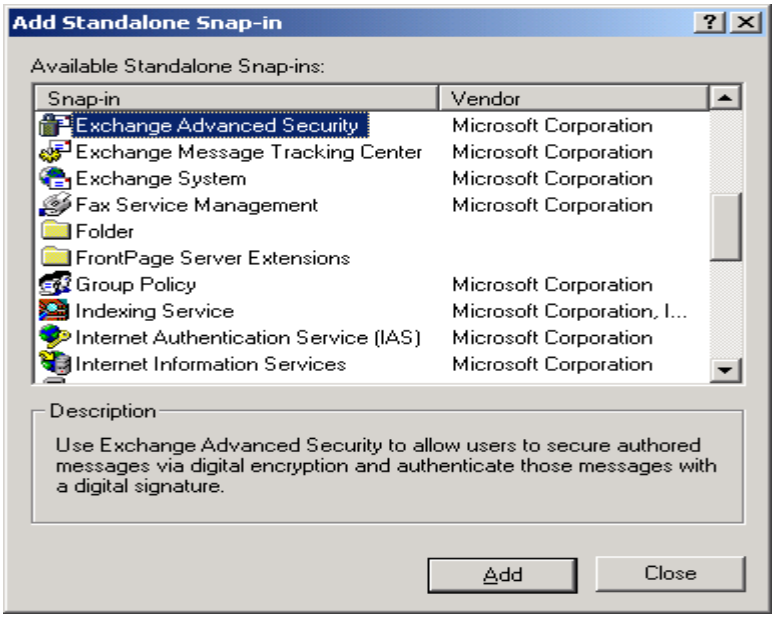
The Exchange 2000 System Manager program allows you to control KMS functions through objects. Each Administrative Group (AG) has it's own Advanced Security Object. In an Exchange 2000 Organization, each Administrative group can host it's own KM Server or point to a KM server in another AG.



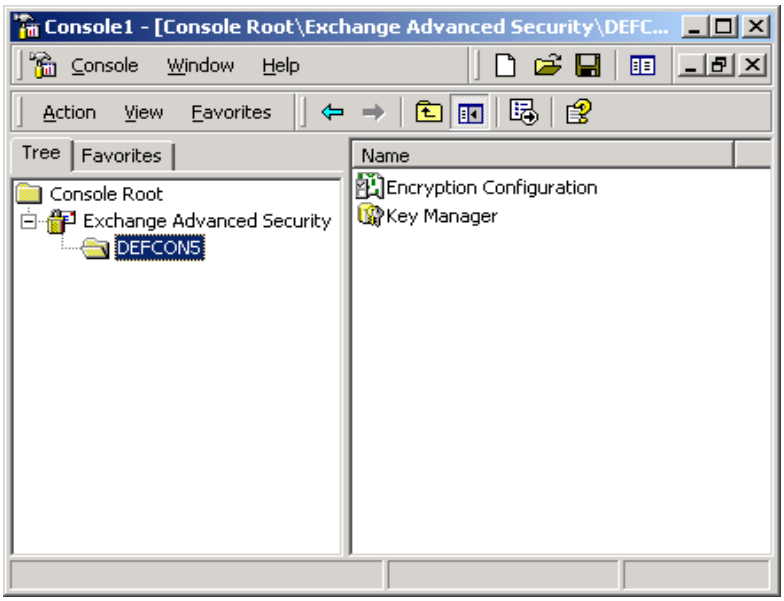
Exchange 2000 has the ability to create a MMC for advanced security

only. This allows administrators to delegate advanced security without having to worry about users accessing other vital information.

1. Open a MMC session
2. Click on **Console** and click **Add/Remove Snap-in** and then click **Add**.



3. Click **Exchange Advanced Security** and then click **Add**.
4. Enter the name of the Domain you want to connect to and click **OK**, then click **Close** and then **OK**.
5. The recipient security tab (which appears when the recipient's properties are displayed) allows you to enroll, revoke, and recover the recipient's keys.



If Things Go Wrong

To back up from bad installation we need to ….yes?

Exchange 2000 KMS Bulk Import/Export

An important feature of KMS in Exchange 2000 Server is the capability to import or export users from one KM server to another. This is available in Exchange200 native mode only.

In Exchange 5.5, if a user was moved to another KMS database, the user would have to decrypt (using SECTOOL) and then re-encrypt once they were moved to the new server.

If you have a large organization with multiple administrative groups and someone is transferred to another department, you might need to move the user to another administrative group. That can entail moving that user's keys to a different KM server. The export and import feature makes it possible for you to move one or more person's keys securely and without risk of misdirection.

Because a user can exist on only one KM server, you must first export a user or group of users before you can import them onto another server.

Exchange 2000 Server incorporates a key export and import feature that enhances the organizational flexibility of organizations that use advanced security, by allowing them to migrate users' key history to a new location. After the key history is migrated to the new location (from one Administrative group to another), the users can access their old keys, and therefore can read e-mail that was encrypted in the old location. KM server keeps these key histories in a database and maps the user to a specific Global Unique Identifier (GUID)

Note: In Exchange 5.5, users were mapped to their DN in the KMS database. When a user leaves an organization (Suki White, DN = White) and the mailbox is deleted, the user's key history remains in the database. When another mailbox is created with the same DN and if they enrolled in advanced security, the user receives not only a new private key but also the Key history of the previous Suki White.

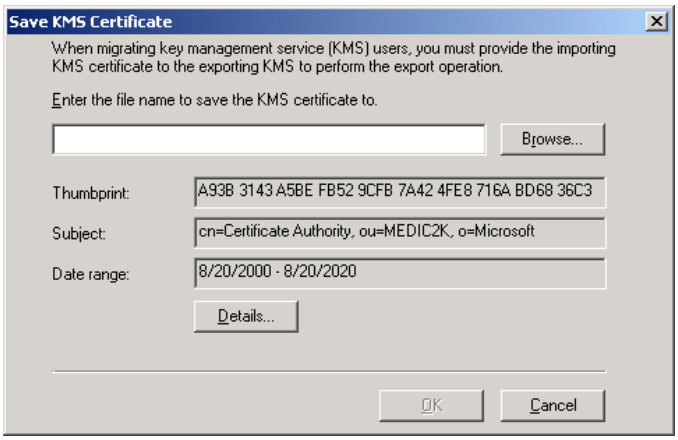
Exchange KMS Key Export/Import Wizard provides clear instructions for exporting and importing users and simplifies the process for administrators. The wizard logs the results of the export and import processes into the *filename*.explog and *filename.*implog files. The wizard exports not only the user's record, but also deletes it from the KMS database (kmsdir.edb).

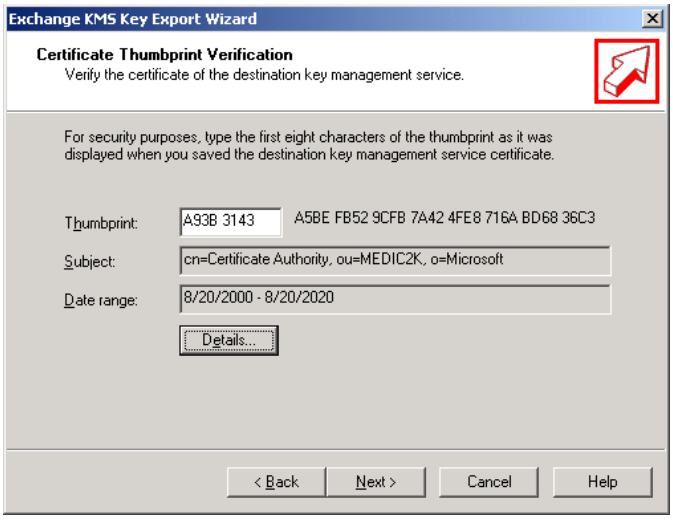
The following security measures are built into the KMS exporting/importing process:

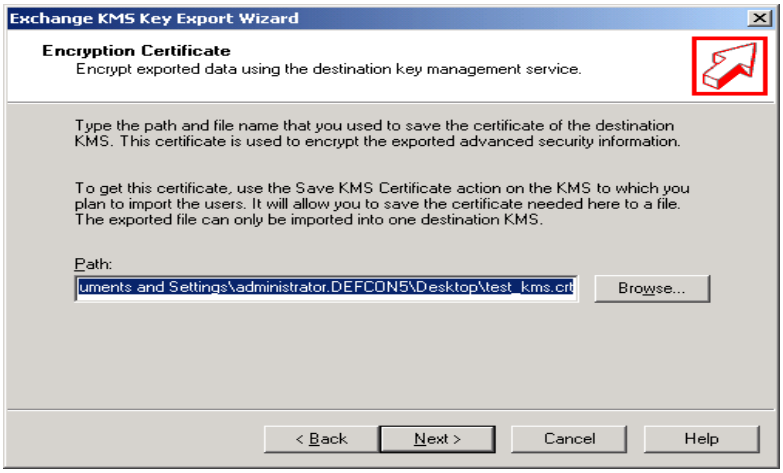
- You cannot move users from a KM server until you specify the destination (importing) KM server.
- Users cannot be imported until they have been exported from another KM server.

- All information is exported in an encrypted file. This exported file can be imported by the designated destination Key Management server only. (To undo your actions, the server that originally exported it can also import the file.)

Before you begin, save the certificate of the destination KM server. This certificate is used to encrypt all exported Advanced Security information.


Save KMS Certificate
When migrating key management service (KMS) users, you must provide the importing KMS certificate to the exporting KMS to perform the export operation.
Enter the file name to save the KMS certificate to.
Browse...
Thumbprint:
A93B 3143 A5BE FB52 9CFB 7A42 4FE8 716A BD68 36C3
Subject:
cn=Certificate Authority, ou=MEDIC2K, o=Microsoft
Date range:
8/20/2000 - 8/20/2020
Details...
OK
Cancel


Exchange KMS Key Export Wizard
Certificate Thumbprint Verification
Verify the certificate of the destination key management service.
For security purposes, type the first eight characters of the thumbprint as it was displayed when you saved the destination key management service certificate.
Thumbprint:
A93B 3143
A5BE FB52 9CFB 7A42 4FE8 716A BD68 36C3
Subject:
cn=Certificate Authority, ou=MEDIC2K, o=Microsoft
Date range:
8/20/2000 - 8/20/2020
Details...
< Back
Next >
Cancel
Help


Exchange KMS Key Export Wizard
Encryption Certificate
Encrypt exported data using the destination key management service.
Type the path and file name that you used to save the certificate of the destination KMS. This certificate is used to encrypt the exported advanced security information.
To get this certificate, use the Save KMS Certificate action on the KMS to which you plan to import the users. It will allow you to save the certificate needed here to a file. The exported file can only be imported into one destination KMS.
Path:
uments and Settings\administrator.DEFCON5\Desktop\test_kms.cri
Browse...
< Back
Next >
Cancel
Help

Preview Documentation

Note: Only users running Outlook 2000 SR1 can be exported.

Archive database details

A user can have multiple entries in the KMS database, each entry representing a layer of certificate history. If old certificates are not maintained, users cannot decipher old enciphered mail. Each record in the database contains up to four certificates, and related state information and certificate data (serial number, expiration date, etc.). Stored certificates include two data encipherment certificates (an X.509v1 certificate generated by KMS and an X.509v3 certificate generated by MS Certificate Server) and two public signing certificates (one for each v1 and v3).

If administrators turn off Exchange 4.0/5.0 backward compatibility, newly generated records will not contain either of the X.509v1 certificates. The signing certificate and the encipherment certificate differ only in that the encipherment private key is enciphered and archived, and the signing private key is generated once and never leaves the client's computer.

Finally, the archive database contains a two-field record for each KMS administrator—Security ID (SID) and password. When an administrator's privileges are revoked, the corresponding SID and password are deleted from the database.

Things to know

Outlook does not use CRL Distribution Points by default. You must set a registry key to allow Outlook to locate a valid CRL when it receives secure email and does not already have a valid CRL for the issuer. Setting the registry key will allow Outlook to use the CRL Distribution Points in the certificate to get a valid CRL for the issuer, providing that the certificate includes a CRL Distribution Point.

Create the following registry key on the client: HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\{7801ebd0-cf4b-11d0-851f-0060979387ea} PolicyFlags=dword:0x00010000

Additional References

Q192044	XADM: Setting Up X509v3 Certificates on Exchange 5.5 SP1 KMS with Local Certificate Server
Q242466	XADM: Error Message: The Key Management Server Is Unable to Operate
Q247814	XADM: Key Management Server Cannot Grant V3 Certificates to Users with Long Distinguished Names
Q218802	XADM: Can't Enroll Using X.509 V3 Certificates

Preview Documentation

Q216653	XADM: Certificate Server is Exclusive to Exchange Advanced Security
Q197965	XGEN: FAQs on High and Low Encryption in Exchange Server
Q216922	Certificate Server Does Not Create Backups of Installed Keys

The latest security information from Microsoft is available at the following Web site:

http://www.microsoft.com/security

Information for software developers is available through the Microsoft Developer Network (MSDN) at the following Web site:

http://msdn.microsoft.com

Additional white papers about Windows 2000 Server and security are available at the following Web site:

http://www.microsoft.com/windows/server/technical/security/default.asp

Secure/Multipurpose Internet Mail Extensions (S/MIME) Internet standards are available at the following Web site:

http://www.imc.org/ietf-smime

U.S. Government

The National Institute of Standards and Technologies (NIST), in conjunction with the Department of Defense (DOD) and the National Security Agency (NSA), has published information on U.S. Government federal public key infrastructure (PKI) efforts. Published documents are available at the following Web site:

http://csrc.nist.gov/pki

The latest information about the Windows 2000 Server operating system is available at the following Web site:

http://www.microsoft.com/windows2000/default.asp

Additional white papers on Windows 2000 Server and Key Management Server Interoperability is available at the following Web site:

http://support.microsoft.com/support/exchange/content/whitepapers/win2k_kms.doc

CHAT SERVICES

Introduction to Chat Services

The Internet Relay Chat (IRC) protocol was developed to support transmission of text in live conferences over the Internet. Typically, this environment consists of a single process on the chat server that provides the focal point for clients and other servers that connect to it. The chat server performs the multiplexing and other tasks required to deliver messages from each connected client to all the others.

Today, many organizations install Chat services in one of the following models:

1. Single Exchange 5.5 Server running One Chat Service for small to medium size organizations
2. Multiple Exchange 5.5 Servers running the same Chat Service for medium to large organizations
3. Multiple Exchange 5.5 Servers running multiple instances of Chat Service typically found at Application Service Providers or Internet Service Providers.

Chat services in Exchange 2000 have been designed to support greater scalability than was available in previous releases. While Exchange 5.5 Chat relied on portals between Chat Servers (up to 255) to add additional concurrent connections to the Chat infrastructure, the Exchange 2000 release can host an equivalent number of users on an individual server. This release allows the consolidation of larger chat channels that were previously hosted on several Exchange 5.5 servers onto one Exchange 2000 server.

Migrating from Exchange 5.5 Chat Service

If you have been using the Chat Service included with Microsoft Exchange Server 5.5, you may have customized some settings or configurations that you want to continue using when you install Microsoft Exchange 2000 Chat Service. The Chat migration tool enables you to create a new Chat community on an Exchange 2000 server with the name of the Exchange 5.5 Chat server, and with most of the same settings and configurations.

In situations where a single server has served the chat community, it is recommended that you perform an in-place upgrade of Exchange 5.5 server (unless existing Exchange 5.5 Server is not meeting minimum hardware requirements for Exchange 2000). For more details on in-place upgrades, please refer to the Exchange Deployment guide and Exchange 2000 Online Help. Note that you must reconfigure the ports so Chat Service 5.5 is not using the default port 6667 or the alternate default port 7000.

Caution: If this step is not completed, Exchange 2000 setup will fail to install Chat Service. However, this step is not necessary if the target is a different machine than the source.

In the case where multiple servers were used, use the migration tool to migrate chat services. There is one caveat: the migration tool will not migrate Chat Service 5.5 extension data. This means that if you had a Profanity Filter or Transcription Extension with your Chat Service 5.5, it is now necessary for you to migrate these extensions manually, so that they will work with Exchange 2000 Chat Service.

The new Chat community you create with the migration tool will be totally separate from your Exchange 5.5 Chat server. If your Exchange 5.5 Chat installation consisted of one Chat service, spanning a network of two or more Exchange servers, be aware that this type of chat installation is not supported by Microsoft Exchange 2000 Server. Using the migration tool, you can create similar Chat communities on each of your Exchange 2000 servers. Each of the Exchange 2000 Chat communities on each of the Exchange 2000 servers in your domain will operate separately, with separate user lists and channel lists.

Chat Service Migration

To migrate to Microsoft Exchange 2000 Chat Service from the previous version:

1. Install Windows 2000 Server on your target machine, creating a new domain, or configuring it as a member server of an existing domain.
2. Install Exchange 2000 Chat Service on each target machine.
3. If you are installing Exchange 2000 on a network of servers on which Chat Service 5.5 spanned two or more servers, upgrade all of the machines to Windows 2000 Server. All the servers must be in the same domain.
4. Run the Chat migration tool (**chatmig.exe**) from the **exchsrvr\bin** directory, and specify the target. The target must be the administration group to which you want to migrate Chat.

After you run the migration tool, a new Chat community is created on the Exchange 2000 server with the name of the 5.5 Chat server, and with similar server, channel, ban and class configurations. Until you are ready to cut over to the Exchange 2000 server, both Chat services will and can run simultaneously.

If you are installing Exchange 2000 Chat Service on a separate machine from the one where Chat Service 5.5 is installed, copy the migration tool from the **exchsrvr\bin** directory on the target machine to the source machine. Run the Chat migration tool, identifying the Chat Service 5.5

directory as the source, and the Exchange 2000 Chat Service directory as the target.

To uninstall Chat Service 5.5 run the uninstaller shipped with Exchange Server 5.5.

If Things Go Wrong

To back up from bad installation we need to …yep, we sure do.

Introduction to Microsoft Exchange Server Scripting and Routing

Using the Scripting Agent and Routing capabilities of Microsoft Exchange Server version 5.5, you can turn a folder into an application, enabling it to run event-triggered server-side scripts and control the flow of messages, forms, or documents among recipients in your organization.

You can use the Microsoft Exchange Server Scripting Agent and Microsoft Exchange Server Routing independently of one another, though the Routing capabilities make use of underlying Scripting Agent technology.

Microsoft Exchange Server Scripting Agent

The Microsoft Exchange Server Scripting Agent lets you use server-side scripts that run as a result of events occurring in folders. Such triggering events include timer events and actions such as posting, editing, or deleting a message. They can take place in private folders (mailboxes) or public folders. You need sufficient folder permissions to install and enable event-driven folder scripts.

You can write these scripts in Microsoft Visual Basic Scripting Edition (VBScript), JavaScript, and other optionally installed scripting engines. You can use Notepad (by default) or Microsoft Visual Studio (specifically, the Microsoft Developer Studio IDE) as your script editor, in conjunction with Microsoft Outlook version 8.03.

Collaboration Data Objects (CDO) (formerly Active Messaging) is the primary interface that scripts use to access and manipulate data in Microsoft Exchange information stores. (These scripts use CDO objects to perform their operations.)

Microsoft Exchange 5.5 Server Routing

From a high-level or architectural viewpoint, the Exchange Server routing solution is a hub-and-spoke architecture. An Exchange Public Folder (or a mailbox) acts as the hub, and the flow of messages to and from each participant in the route appears as the spokes.

From an implementation perspective, the Exchange Server routing

solution can be conceived of as a stateful scripting agent with four components:

- A routing engine, which manages the state and state transitions of the route.
- A routing map or state diagram, which describes the process logic to be used in the route.
- A set of objects to assist in manipulation of the map and other items.
- Some scripts, which describe the set of functions that will be used in execution of the route.

The Routing Engine has been implemented as a custom agent that runs under the control of the Exchange Event Service. It is configured on a per-folder basis and responds to all the events generated by the Event Service on that folder.

Event Support in Exchange 2000

Exchange 2000 provides a comprehensive server event model, which enables designers to create many types of applications that were previously impossible. Two kinds of events exist: transport events and Web Storage System events (both synchronous and asynchronous).

Transport events are used to track and customize the basic operations of Exchange 2000 while the system processes SMTP and NNTP requests. For example, an application could use transport events to attach a corporate disclaimer on all outgoing mail, rewrite the mail headers, or force all mail sent to a distribution list with more than 1,000 addresses to go through an approval process.

Web Storage System events are OLE DB–based server events that provide an Application Programming Interface (API) through which developers can extend the features of Web Storage System. Web Storage System events include synchronous, asynchronous and system events. Synchronous events occur before the change is committed to the store. Asynchronous events are essentially notifications that happen after the item is saved or deleted. There are three Web Storage System events: starting the Web Storage System, stopping the Web Storage System, and the timer.

Synchronous events are new to Exchange 2000. When a synchronous event is triggered, Exchange performs no further processing on an item until the event’s associated business logic is executed. For example, a synchronous event could be associated with a folder that checks the validity of the digital signature of an expense report before the expense report is sent to management for approval. Events allow applications to define custom actions on particular events. For example, a workflow application can initiate a workflow process when an object is saved to a

SUMMARY

particular folder.

You can control which users can register event sinks and what the event sinks can access. For more information about Event Security, go to http://www.microsoft.com/exchange.

Upgrading your Event-Script Services

Exchange 2000 runs an Exchange 5.5 compatible event–script service. Event scripts run with no modification. Because security is stronger in Exchange 2000, certain permissions in the scripts might need to be changed. Because event scripting in Exchange 2000 is more powerful, you may need to rewrite scripts to take advantage of Exchange 2000 functionality.

If Things Go Wrong

To back up from bad installation we need to …again?

Introduction

It is unlikely that your existing third–party tools, connectors, virus checkers, and so on will work correctly after you upgrade an Exchange 5.5 server to Exchange 2000. Most third–party applications use the Directory Application Programming Interface (DAPI) protocol to access the Exchange 5.5 directory. This is not supported within Active Directory. Applications must be rewritten to use ADSI or LDAP to manipulate Active Directory. Either disable or remove third–party software prior to upgrade

Before you do your final upgrade:

- Make complete backups of the servers so you can recover if something goes wrong
- Create a lab where you will test your new setup
- Run test scenarios
- Plan for outage (in off-peak hours and weekends)
- Notify all users for the expected outage. Add a 30% buffer to the time you think the outage will last.

If Things Go Wrong

To back up from bad installation we need to ….

This chapter gave you information on the upgrade processes. For more

information and for complex migrations, you should contact Microsoft Consulting Services (MCS). Microsoft Consulting Services (MCS) has many experienced consultants who can carry out the migration process described. For more information see

http://www.microsoft.com/ClO/services/consulting.htm

Many Microsoft Certified Solution Providers (MCSPs) also have the skills needed to successfully complete a project of this nature and can be contacted through http://www.microsoft.com/mcsp.

Key Management Services

CAST—a variable key length encipherment algorithm developed by Carlisle Adams and Stafford Tavares of Northern Telecom Research. This algorithm supports keys 40 to 128 bits long.

Clear item—a message that is readable when accessed by..., that is not enciphered.

Data Encipherment Standard—(DES/Triple DES) an IBM symmetric encipherment block cipher that uses a fixed 56-bit key. It was defined and endorsed by the U.S. government in 1977 as an official standard and is regarded as the most widely used cryptosystem in the world. A process of enciphering plain text three times with DES and three different potential series of actions. DES-EEE3 (three DES encipherments with three different keys), DES-EDE3 (three DES operations in the sequence *encipher-decipher-encipher* with three different keys), or DES-EEE2 and DES-EDE2 (same as the previous formats except that the first and third operations use the same key).

MD (Message Digest Algorithm, MD4, MD5)—developed by Rivest, this takes a message of arbitrary length and produces a 128-bit message digest. The algorithm is optimized for 32-bit machines. Description and source code for MD4 and MD5 can be found as Internet RFCs 1319 – 1321.

Opaque item—message text that cannot be read without being deciphered, also known as an enciphered item

RC2—a variable key-size block cipher designed by Rivest for RSA Data Security. “RC” stands for “Ron’s Code” or “Rivest’s Cipher.” Faster than DES and is designed as a “drop-in” replacement for DES. Because of the variable key size it can be made more secure or less secure than DES against exhaustive key search. The algorithm is confidential and proprietary to RSA Data Security.

RC4—a variable key-size stream cipher with byte-oriented operations designed by Rivest for RSA Data Security.

SASL—Simple Authentication and Security Layer. Defined in RFC2222.

Secure Hash Algorithm version 1—Published in 1994 as a federal information-processing standard (FIPS PUB 180), this was developed by NIST. Similar to the MD4 family of hash functions, this takes a message of less than 2^{64} bits and produces a 160-bit message digest. It is slightly slower than MD5 but is more secure against brute-force collision and inversion attacks.

SSL—version 3.0 is defined in the Internet draft <draft-ietf-tls-ssl-version3-00.txt>.

X.509—a standard released by the International Telecommunications Union that specifies the formatting of a mechanism to verify public keys issued to security principals in an organization.

Encryption

Encryption is the mathematical transformation of data from a readable, clear text form, into an unreadable, cipher text form. The transformation generally requires additional secret information available only to the sender and intended recipient. This information is called a *key*. The key allows the message to be encrypted by the sender, and decrypted only by the intended recipient using the recipient's private key. Decryption is the opposite of encryption — it transforms unreadable, cipher text data back into readable, clear text form.

Using cryptography provides not only privacy, but it provides an identity every time a user logs on to a network, accesses voice mail, or uses a user name and password to access anything. This identification is called *authentication*. Authentication is a crucial part of network data security.

As electronic transaction use increases over networks, it is important to sign documents electronically. Cryptography provides the ability to create digital signatures, which in many cases are as legally binding as written ones.

Hash Functions

A hash function provides a means of computing an electronic fingerprint, or checksum of a message. This electronic fingerprint is called the *hash* of a message.

Hashing secures messages and private key data by using them as elements in a mathematical function that creates a checksum of the package. The algorithm is then used on the receiving end to decrypt the message. Hashes typically compute quickly, and are designed so that every imaginable message can have a unique hash. Hash algorithms include MD-4, MD-5, and SHA-1.

Ciphers

A cipher is a mathematical function for encrypting and decrypting data. It

is performed on readable clear text data to convert it to an unreadable version called cipher text. There are four types of ciphers: symmetric, asymmetric, block, and stream.

Symmetric cipher Symmetric, or shared-key, ciphers are a form of data encryption in which a single key, known by the sender and the recipient, is used to encrypt and decrypt a message. While this form of encryption is efficient and effective, it is often difficult to share the key between both parties in a secure manner. It requires that the sender communicate the key to the recipient in a secure way.

Asymmetric cipher The asymmetric cipher, or public-key cipher, is a means of solving the key management problem of symmetric key encryption. This system involves using two keys, one for encryption, and the other for decryption. One of the keys is called the public key, and the other is called the private key. You can use either the public or private key for encryption, and you use the opposite key for decryption. The public key is placed in a directory, or a location available to other users, but the private key is kept in a secure location, and is available only to the owner of the key pair. By using an asymmetric cipher, the sender and recipient do not need to agree on a key before sending data.

Block cipher A block cipher uses shared-key encryption. It takes a message and breaks it into fixed length blocks, and applies the shared-key to each block. In most cases, this block size is 64 bits. The decryption operation takes the encrypted blocks, decrypts each with the same shared-key, and rebuilds the original message.

Stream cipher A stream cipher is another use of symmetric encryption. Stream ciphers process small units of plaintext, usually bits. Stream ciphers are much faster than block ciphers, and can be applied to data as it is sent or received. You do not need to know the size of the message, or receive the entire message before beginning to decrypt the message. This is useful for encrypted conversations over a network such as SSL rather than individually–encrypted messages.

Secure Transmission

Exchange version 5.5 supports authentication of SMTP delivery between servers in three ways: a combination of SASL and AUTH, a combination of SASL/AUTH over SSL, or via Windows NT Challenge/Response and NTRPC.

SASL/AUTH with SSL Enciphered Connection (No Item Encipherment/Signing)

SASL provides a simple base level of authentication and optionally allows two communicating hosts to negotiate a security protocol. Exchange 5.5 supports SASL as an authentication mechanism between SMTP mail transfer hosts, and AUTH (an SMTP extension command), which similarly allows for the authentication of the connection between two mail hosts.

When combined with SSL, these authentication and session security techniques provide an enciphered and authenticated connection between the transfer hosts for the “clear” SMTP transfer.

NT Challenge/Response and NTRPC Connection (No Item Encipherment/Signing)

NT Challenge/Response is an enciphered authentication protocol used between Windows NT servers. If NTRPC encipherment is turned on (the default state between servers) the communication stream between the Windows NT servers is enciphered with 128-bit strength RC-4. This means that the SMTP mail transfer is enciphered, even though the underlying SMTP transfer format is clear text. The items are secure while being transferred over the Internet and only “clear” after they have been received by the authenticated destination server.

X.509v1 Key Exchange (Item Encipherment and Signing)

Exchange versions 4.0, 5.0, and 5.5 support the issuance of X.509v1 certificates to recipients in the Exchange organization. Using a supplied form with Exchange (the certificate exchange form), users in one Exchange organization can send their certificates to users in another Exchange organization. After certificates are swapped, the users can send and verify messages that are digitally enciphered, digitally signed, or both.

X.509v3 Key Exchange (Item Encipherment and Signing)

Exchange 2000 supports the issuance of X.509v3 certificates to recipients in the Exchange organization. Using a form supplied with Exchange, users in one Exchange organization can send their certificate to users in another Exchange organization. After the certificates are swapped, the users can send and verify messages that are digitally enciphered, digitally signed or both.

Certificate Services and the Key Management Service

Exchange 2000 KMS uses Certificate Services to provide security on the application layer of the messaging system. It produces X.509v3 user certificates that Outlook 2000 uses for digital signature and encryption. The X.509v3 user certificates are recognized by Secure/Multipurpose Internet Mail Extension (S/MIME) clients, and ensure interoperability among different clients when used across the Internet.

Note The Enroll Agent (Computer), Exchange User, and Exchange User Signature templates of the CA must be enabled before KMS installs**.**

KMS can be configured to require more than one administrator to be present before performing each KMS task. Each task is initially configured to require only one password, and the first password is always “password” until it is changed.

KMS implements a dual-key architecture, which gives each enrolled user two key pairs and certificates, one used for encryption, and the other used for signing. KMS uses this architecture so the signing key does not need to be archived in KMS and enforces non-repudiation.

Earlier versions of KMS restricted KMS to just one server per site. However, Exchange 2000 KMS fits the site-based model. The KMS organizational unit becomes the Administrator Group and each new administrator group created allows KMS to install on or point to an existing KMS server. You can also enroll users in bulk by enrolling selected users, groups, or servers.

In addition to taking advantage of the built-in security of Windows 2000 Server, Exchange 2000 Server provides advanced security in the form of an optional component named the Key Management Service (KMS). The KMS works in conjunction with the Active Directory, directory services to manage the encryption keys used to encrypt e-mail messages. It integrates with Microsoft Certificate Services to provide services.

Exchange System Manager—manages the Advanced Security operations for an organization, including issuance and revocation of certificates, recovery of certificates, setting of site encipherment algorithm preferences.

KMS Archive database—stores private encrypting keys

Windows 2000 Certificate Server—Windows 2000 Certificate server is required to run KMS

Active Directory— Generates private signing and encrypting keys and stores certificates/preferences for client encipherment and signing. For example, clients state in the directory the maximum strength security that they support (for example, 128-bit encryption). Others clients use this information to secure messages in ways the clients will be able to decipher.

Client (Outlook)—the client interface to Advanced Security. It reads, from the Active Directory users' certificates, performs all cryptography on mail messages, and protects private keys in a secure store

Event Script Services

Event Service – A Windows NT service that is installed by Exchange server 5.5. This service allows programmers to write programs that use Exchange's Event Handler to process events that occur in a Public Folder or Mailbox.

Exchange Server Scripting Agent —Exchange Server Scripting Agent lets you use server-side scripts that run as a result of events occurring in Exchange folders. There are 4 events that can trigger the scripting agent, timer events and actions such as posting, editing, or deleting a message.

Routing Objects—Component Object Model (COM) objects that are used to program Exchange's routing engine behavior. These objects allow the creation and manipulation of *process maps*, which define the series of states to be tracked by the routing engine and the activities to be performed at each step. Routing Objects are used primarily in workflow applications.

MAPI—(Messaging Application Programming Interface) A messaging architecture API enabling multiple applications to interact with multiple messaging systems across a variety of hardware platforms. MAPI is built on the COM foundation.

CDO—Client & Server **-** (Collaboration Data Objects) An object model for building messaging or collaboration applications. In versions previous to 1.1, CDO was called OLE Messaging; in version 1.1 it was called Active Messaging. It is designed to simplify the creation of applications with messaging functionality, or to add messaging functionality to existing applications.

ADO—(Active Data Objects) A programming interface. ADO provides access to data, such as a front-end database client or middle-tier business object using an application, tool, language, or even an Internet browser. ADO is the interface you needed for 1- to n-tier client/server and Web-based data-driven solution development. ADO is designed as an easy-to-use application level interface to, OLE DB. OLE DB provides access to sources, including relational and non-relational databases, email and file systems, text and graphics, and custom business objects.

ADSI—(Active Directory Services Interface). A directory service abstraction interface that allows programming languages that are compatible with the Component Object Model (COM), such as Visual Basic, VBScript, JavaScript, C, and C++ to make common directory calls to an underlying directory service. ADSI providers include Lightweight Directory Access Protocol (LDAP), NDS, Bindery, and Windows NT (SAM). Programmers and system administrators normally use ADSI to automate or script the bulk manipulation of directory entries.

ASP—(Active Server Pages). An open, compile-free application-scripting environment, in which you can combine HTML pages, scripts, and Microsoft ActiveX server components to create Web-based applications. Active Server Pages offers native support for Microsoft Visual Basic Scripting Edition (VBScript) and Microsoft JScript, and supports other scripting languages such as REXX, Python, and Perl through Active Scripting plug-ins. Active Server Pages also supports ActiveX Scripting, allowing virtually any scripting engine to be used. It allows Web developers to write scripts that are executed on either the server or the client.

The Exchange Event Scripting Agent

Microsoft Exchange Event Service (events.exe)—A Microsoft Windows NT service that runs on the Microsoft Exchange Server computer. This service is installed as part of the Microsoft Exchange Server 5.5 setup. Using Microsoft Outlook, you configure this service to monitor events that occur in folders you specify.

Folders—The folders upon which event scripting works can be Microsoft Exchange private folders (mailboxes) or public folders. These folders must be stored on the Exchange Server in order for events to occur. Locally stored folders such as .pst files cannot take advantage of the Exchange Scripting Agent. Examples include Calendar, Contact, or Inbox folders.

Microsoft Exchange Scripting Agent—An ActiveX™ scripting engine that processes scripts. This element is a COM object that exposes the **IExchangeEventHandler** interface.

User-created scripts—These scripts, which call CDO objects/interfaces, constitute the actual CDO logic that works on folders. Scripts are stored as messages in the folder and as such, travel with the public folder application. These scripts are visible only through the **Agents** tab in Microsoft Outlook or through a MAPI utility such as MDBVU32.EXE.

Microsoft Exchange Server Routing

Routing engine—This COM component runs on the Event Service on Microsoft Exchange Server version 5.5. It acts as a simple state engine that executes and tracks multiple process instances within a Microsoft Exchange folder. The state is advanced when events fire within the folder. The routing engine supports the execution of flow-control activities (workflow) directly, and it can call VBScript functions for other activities. Microsoft Exchange Server Routing also works with the Microsoft Transaction Server (MTS).

Routing Objects—These COM objects are used to program the routing engine's behavior. These objects allow the creation and manipulation of

process maps, which define the series of states to be tracked by the routing engine and the activities to be performed at each step.

VBScript Actions—These are a default set of common VBScript functions for routing. They include functional activities such as Send, Receive, and Consolidate as well as evaluation activities such as IsTimeout and IsNDR. Developers can extend these activities with custom script functions as their application needs dictate.

Using these components, developers can create applications such as simple process designers, which allow users to define and create the steps in routing/approval applications. An example of such a program, the Microsoft Exchange Routing Wizard, is included with Microsoft Exchange 5.5 routing objects as a sample application.